



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/675,116	09/28/2000	Grzegorz J. Czajkowski	SUN-P5075-RSH	9136
22835	7590	02/26/2004	EXAMINER /	
PARK, VAUGHAN & FLEMING LLP 508 SECOND STREET SUITE 201 DAVIS, CA 95616			ALI, SYED J	
			ART UNIT	PAPER NUMBER
			2127	4

DATE MAILED: 02/26/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/675,116

Applicant(s)

CZAJKOWSKI ET AL.

Examiner

Syed J Ali

Art Unit

2127

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 08 December 2003.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-21 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-21 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This office action is in response to Amendment A, paper number 3, which was filed December 8, 2003. Claims 1-21 are presented for examination.

2. The text of those sections of Title 35, U.S. code not included in this office action can be found in a prior office action.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 1-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gosling (previously cited) in view of Jagannathan et al. ((USPN 6,496,871) (hereinafter Jagannathan).

As per claim 1, Gosling discloses a method for verifying type safety of an application snapshot, the application snapshot including a state of an executing program that is moved from a first computing device to a second computing device across a network in order to continue execution on the second computing device, the method comprising:

receiving the application snapshot from the first computing device on the second computing device, wherein the application snapshot includes a subprogram, an operand stack, and a point of execution (col. 6 lines 28-46, "The stack snapshot storage structure 346 is

Art Unit: 2127

bifurcated into a directory portion 348 and a snapshot storage portion 350. The directory portion 348 is used to store target instruction identifiers [e.g., the absolute or relative address of each target instruction] while the snapshot portion 350 is used to store virtual stack 344 snapshots associated with the target instruction identifiers”);

examining the application snapshot to identify the subprogram and the point of execution within the subprogram (col. 7 lines 31-44, “A first pass is made through the bytecode program in order to extract target information associated with conditional and un-conditional jumps and loop instructions....For instance, the absolute or relative address of the target instruction may be stored in the next available slot of the directory”);

examining the subprogram to determine an expected structure of the operand stack at the point of execution (col. 7 line 58-65, “a second pass through the bytecode program is initiated in order to verify proper use of the operand stack and of data types by the bytecode program”);

validating that the state of the application snapshot on the second computing device is consistent with the expected structure of the operand stack (col. 7 line 58-65, “a second pass through the bytecode program is initiated in order to verify proper use of the operand stack and of data types by the bytecode program”); and

if the state of the application snapshot is validated as consistent with the expected structure of the operand stack, resuming execution of the application snapshot on the second computing device (col. 10 lines 56-64, “If no more instructions are to be processed, then the verifier will then set a verification status value 245 for the program to True (544), signaling the completion of the verification process”).

Jagannathan discloses the following limitations not shown by Gosling, specifically that the application snapshot contains dynamic variables and defines the dynamic state of the executing program (col. 11 lines 6-33, "Tasks and data may freely and dynamically migrate among the machines in the network associated with creating their agent", "When an agent consists of multiple threads executing on different bases and an agent migration method is called, all of these threads are preferably gathered at one of the bases before migrating to the target base. Object migration permits internal data and associated threads to migrate").

It would have been obvious to one of ordinary skill in the art to combine Gosling with Jagannathan since the disclosure of Gosling, while providing a sound means for verifying an application before execution on another machine, requires that the verification procedure be performed before execution of the application. This prevents efficient use of a network system, especially one that is dynamically configurable. Specifically, load balancing mechanisms are known that redistribute executing entities across a network as new machines are added or removed in order to avoid bottlenecks and starvation situations. By implementing a system that precludes migration of ongoing applications, these load balancing mechanisms cannot be used. Jagannathan provides a system that allows dynamic process migration while also maintaining state information related to the ongoing execution. Jagannathan acknowledges that the problem of migration of ongoing processes has been addressed, but the prior art does not allow for state information related to those processes to be easily migrated (col. 4 lines 47-59, "Telescript and Odyssey agents do not allow distributed state: when an agent moves, it is necessary that its entire state moves along with it. This limitation significantly reduces functionality and efficiency. In the case of Odyssey, only the state as found in the heap can move - the state of the stack,

Art Unit: 2127

program counter, and registers are all lost. Telescript imposes similar restrictions"). Thus, Jagannathan seeks to improve the prior art by not only providing a system that allows migration of ongoing processes, but also allows the state information to be migrated (col. 5 lines 28-35, "there remains a need for a distributed computing system which is easy to program and which...allows easy and efficient process migration, in whole or in part, among distinct machines"). Thus, the combination of Gosling and Jagannathan would provide an exemplary model for verifying the type safety of an executing application, while maintaining state information related to the application as it is migrated from one machine to another.

As per claim 2, Gosling discloses the method of claim 1, wherein examining the subprogram to determine the expected structure of the operand stack at the point of execution involves examining the subprogram with a code verifier, wherein the code verifier ensures that:

the subprogram does not cause the operand stack to overflow and underflow (col. 8 line 46 - col. 9 line 18, "If the operand stack has insufficient data [452] for the current instruction, that is called a stack underflow, in which case an error signal or message is generated", "If the operand stack has insufficient room to store the data to be pushed onto the stack by the current instruction [472], that is called a stack overflow, in which case an error signal or message is generated");

a use of a local variable does not violate type safety (col. 10 lines 16-27, "If the currently selected instruction reads data from a local variable [510], the verifier will compare [512] the data type code information previously stored in the corresponding local variable with the data type requirements"); and

Art Unit: 2127

an argument of an instruction is of an expected type (col. 6 lines 6-13, “for each datum that would be popped off the stack for processing by a bytecode instruction, the verifier pops off the same number of data type value off the virtual stack 342 and compares the data type values with the data type requirements of the bytecode”, wherein the datum that is popped off the stack includes operands as well as any operations and arguments associated with the instruction).

As per claim 3, Gosling discloses the method of claim 1, wherein the operand stack contains at least one local variable, at least one argument that is passed as a parameter to the subprogram, and an offset to the point of execution within the subprogram (col. 5 lines 21-29, “the verifier 240 uses a stack counter 342, a virtual stack 344, a virtual local variable array 345, and a stack snapshot storage structure 346”; col. 6 lines 6-13, “for each datum that would be popped off the stack for processing by a bytecode instruction, the verifier pops off the same number of data type value off the virtual stack 342 and compares the data type values with the data type requirements of the bytecode”; col. 6 lines 28-46, “The directory portion 348 is used to store target instruction identifiers [e.g., the absolute or relative address of each target instruction]”).

As per claim 4, Gosling discloses the method of claim 2, wherein the expected structure of the operand stack includes a collective size of entries and the types of entries expected on the operand stack at the point of execution within the subprogram (col. 7 lines 20-30, “The verifier 300 creates [402] the virtual stack 344 and creates the virtual local variable array 345 by designating arrays of locations in memory to store operand and local variable data type

Art Unit: 2127

information...[and] designates [406] a register to act as a stack counter 342 for keeping track of the number of virtual stack entries”).

As per claim 5, Gosling discloses the method of claim 1, further comprising restoring the state of an object within the application snapshot on the second computing device by changing a pointer from an address of the object on the first computing device to an address of the object on the second computing device (col. 6 line 62 - col. 7 line 9, “The bytecode program 350 includes instructions for stack manipulations 352 and 354 [push integer onto the stack and pop integer from the stack respectively], a forward jump 356 and its associated target 364, a backwards jump 366 and its associated target 362, and a do loop 358 and its associated end 360 [which may be an unconditional or conditional branch instruction, depending on the type of do loop]”, wherein when the bytecode program is downloaded, verified, and set to run on the target machine, the target address of the instructions are changed to correspond to the new machine on which it will be executed).

As per claim 6, Gosling discloses the method of claim 4, wherein validating that the state of the application snapshot on the second computing device is consistent with the expected structure of the operand stack involves ensuring that the collective size of entries and the types of entries on the operand stack agree with the collective size of entries and the types of entries expected on the operand stack (col. 9 lines 44-62, “In the preferred embodiment, a mismatch will arise if the current virtual stack and snapshot do not contain the same number or types of entries.

Art Unit: 2127

Furthermore, a mismatch will arise if one or more data type values in the current virtual stack do not match corresponding data type values in the snapshot”).

As per claim 7, Gosling discloses the method of claim 1, wherein resuming execution of the application snapshot involves restarting the subprogram at the point of execution within the second computing device (col. 6 line 62 - col. 7 line 9, “The bytecode program 350 includes instructions for stack manipulations 352 and 354 [push integer onto the stack and pop integer from the stack respectively], a forward jump 356 and its associated target 364, a backwards jump 366 and its associated target 362, and a do loop 358 and its associated end 360 [which may be an unconditional or conditional branch instruction, depending on the type of do loop]”, wherein the associated target instruction is loaded on the second machine, and the execution of the bytecode program is resumed at said target instruction).

As per claims 8-14, Gosling discloses a computer-readable storage medium storing instructions that when executed by a computer causes the computer to perform the method of claims 1-7, respectively (Fig. 2, wherein a client/server model is disclosed for implementing the verification method of claims 1-7, respectively. Furthermore, a computer-readable storage medium is part of both the client and server systems, and would store the bytecode programs to be verified, as well as the verification module). Since the remainder of the limitations are similar to those presented in claims 1-7, the discussion of claims 1-7 above forms the basis for rejection of the present claims as well.

Art Unit: 2127

It would have been obvious to one of ordinary skill in the art to combine Gosling with Jagannathan for reasons discussed above in reference to claim 1.

As per claim 15-21, Gosling discloses an apparatus that implements the method of claims 1-7, respectively (Fig. 2, wherein the client/sever model disclosed is an apparatus that stores the bytecode programs to be verified, as well as the verification module). Since the remainder of the limitations are similar to those presented in claims 1-7, the discussion of claims 1-7 above forms the basis for rejection of the present claims as well.

It would have been obvious to one of ordinary skill in the art to combine Gosling with Jagannathan for reasons discussed above in reference to claim 1.

Response to Arguments

5. Applicant's arguments with respect to claims 1-21 have been considered but are moot in view of the new grounds of rejection.

Conclusion

6. Applicant's amendment necessitated the new grounds of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after

Art Unit: 2127

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (703) 305-8106. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai T An can be reached on (703) 305-9678. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Syed Ali
February 19, 2004



MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100